Selective sampling importance resampling particle filter tracking with multibag subspace restoration

Jenkins, Mark David; Barrie, Peter; Buggy, Tom; Morison, Gordon

Published in:
IEEE Transactions on Cybernetics

DOI:
10.1109/TCYB.2016.2631660

Publication date:
2018

Document Version
Author accepted manuscript

Selective Sampling Importance Re-Sampling Particle Filter Tracking with Multi-Bag Subspace Restoration

Mark David Jenkins, Peter Barrie, Tom Buggy, Gordon Morison

***Abstract*—The focus of this paper is a novel object tracking algorithm which combines an incrementally updated subspace based appearance model, reconstruction error likelihood function and a two stage Selective Sampling Importance Re-Sampling particle filter with motion estimation through autoregressive filtering techniques. The primary contribution of this work is the use of multiple bags of subspaces with which we aim to tackle the issue of appearance model update. The use of a multi-bag approach allows our algorithm to revert to a previously successful appearance model in the event that the primary model fails. The aim of this is to eliminate tracker drift by undoing updates to the model that lead to error accumulation and to re-detect targets after periods of occlusion by removing the subspace updates carried out during the period of occlusion. We compare our algorithm with several state of the art methods and test on a range of challenging, publicly available image sequences. Our findings indicate a significant robustness to drift and occlusion as a result of our multi-bag approach and results show that our algorithm competes well with current state of the art algorithms.**

***Index Terms*—Object Tracking, Particle Filter, SSIR, Appearance Model**

I. Introduction

VISUAL object tracking is a vital component in the world of computer vision and machine intelligence. Several challenges become apparent during the development of such a system. Object appearance changes through events such as pose variation, object rotation, scale changes, changes in illumination and motion blur, all of which are inherent in video sequences due to their dynamic nature, present a challenge to tracking algorithms. These changes in appearance, in combination with occlusion and background clutter make successful object tracking a highly complex and demanding task. Despite the considerable amount of research carried out in this area in recent years, there has yet to be the development of a definitive solution which is capable of handling all of the aforementioned challenges. A complete review of current object tracking methods is beyond the scope of this paper and a much more in-depth background into the various methods of object tracking can be found in the surveys by Yilmaz *et al.* [1], Cannons [2] and Yang *et al.* [3].

Applications of visual object tracking are vast and cover a multitude of fields. Tasks such as video surveillance [4], human-machine interaction [5], [6], automated vehicle control [7] and human behaviour analysis [8] all benefit from vision systems and utilise object tracking heavily. Methods of visual object tracking fall into one of two categories; generative and discriminative. Discriminative methods [9]–[12] generally require large training sets as they aim to tackle the tracking problem by utilising a classifier to separate the target object from the background or non-target objects. On the other hand, generative methods [13]–[15] aim to find the target object though comparison of an image region against a specific appearance model. We will focus more on the generative methods in this paper.

The particle filter is utilised extensively in visual object tracking [16]–[19]. The algorithm proposed in this paper utilises a selective sampling importance re-sampling (SSIR) particle filter framework in combination with a motion estimation algorithm which aims to maximise the probability of sampling in region of the image most likely containing the target. Our basic appearance model utilises a set of principal eigenvectors which is common among particle filter tracking algorithms [16], [20]. We employ a reconstruction error based detection method and target censorship system utilising the error variance which aims to evaluate the strength of a target match and allow for poor matches to be ignored. The primary contribution of this work is the way in which we construct our appearance model as a two stage bag of subspaces and update in such a way that the algorithm is capable of restoring the appearance model to a previous state should the primary model become polluted with offset or occluded updates.

The remainder of this paper is presented as follows;

- Section II - work related to this research and a rationale for several key algorithm attributes
- Section III - details the components which result in our final algorithm including the major contribution of our work which is the use of multiple subspace bags for appearance model restoration
- Section IV - performance evaluation of our algorithm along with 14 other state of the art algorithms
- Section V - in depth discussion of the algorithm performance on a selection of image sequences

II. Related Work

Generative tracking methods can be broadly described in three main stages; some model of the target is generated, the best match to this model is located in the subsequent frame and the model is updated to adapt to changes. Appearance models can take a number of forms such as correlation filters [21]–[23], subspaces [14], [16], [24] or intensity models [25]–[27]. A combination of representations is also common as in [28]

where Danelljan *et al.* aim to improve the effectiveness of the correlation filter model through the incorporation of colour.

Regardless of the appearance model utilised, an effective update scheme must be implemented to allow the algorithm to adapt to changes in the target. There is considerable debate as to the most effective method of updating the appearance model and each method comes with its own set of problems. The most naive of these update methods is to continuously replace the model after each frame with the newly estimated target. Matthews *et al.* [29] address the fundamental issue with sequential update which is that of tracker drift. Naive sequential update inevitably leads to error accumulation resulting in a steady increase in drift away from the target. The proposed solution to this is to retain some quantity of the target appearance in the initial frame and incorporate this into the appearance model during the update. This technique can be found in some form in many of the generative tracking methods [16], [23], [30] and has shown a significant decrease in tracker drift. Appearance model update presents a second challenge; when or when not to update. Updating the appearance in the event of occlusion can cause the model to learn false data which negatively impacts tracking performance. Some algorithms attempt to detect occlusion as it is happening and postpone updates during this time [31], [32] while others look to update the portion of the appearance model which is not occluded and leave the occluded section unchanged [33], [34]. A third method is to allow constant updates but to provide the algorithm with the option of reverting to an older appearance model which has not been contaminated or undo these updates [24], [35]. We employ a method similar to this but with a two independent bags of subspaces in combination with the possibility of generating new subspaces which will be explored fully later in this paper.

It is highly computationally inefficient to search an entire image on each frame when tracking an object and probability dictates that the target is most likely to be within some radius of its previous location. As a result, multiple methods of searching for possible target locations have been formulated. The grid search method employed by Babenko *et al.* [36] for example, centres its search region around the previous target location and then performs an exhaustive search of a specified radius to find its target. Correlation filters theoretically perform the same operation but more efficiently by exploiting frequency domain techniques [21]–[23] to analyse the entire search region in one operation. Another popular technique is the particle filter. This technique involves random sampling of the region surrounding the previous target location but taking into account the prior probability of that location containing the target. This is usually done by sampling with replacement which encourages higher weighted locations to be sampled multiple times and thus increasing their probability. The commonality with these methods is the starting point of any sort of search. Each method centres its search at the location of the target in the previous frame which may not be the most effective location, especially in the instance of a fast moving target. For this reason we employ a particle filter with a motion estimation stage which predicts the next target location based on its previous trajectory and centres the particle filter on this new location to provide a higher likelihood of locating the target [37], [38].

III. Proposed Algorithm

In this paper we utilise a multi-stage particle filter framework. Stage one of this particle filter is a standard sample importance re-sampling (SIR) particle filter which re-samples based on the weights of the samples in the previous frame while the second stage utilises a linear autoregressive (AR) filter to predict the object position based on its previous trajectory using the Burg method [39] to target sampling towards the most probable object location. We evaluate the likelihood of an image patch containing the target by minimising the error between the patch and its reconstruction from the appearance model basis [24]. Even the most robust form of appearance model will degrade rapidly over time without an appropriate update strategy and as a result, the method by which we handle the model update problem forms the main contribution of this work. We utilise two bags of subspaces, one updated with regards to tracking importance and the other updated temporally to allow for appearance model restoration should the current model fail due to drift or occluded updates.

The remainder of this discussion is divided into several sections covering the main components of our algorithm. *Section A* covers the basic particle filter framework utilised while *Section B* presents further details on the second stage of the particle filter utilising the autoregressive motion estimation. *Section C* describes the form of appearance model used. *Section D* discusses the method used to determine the target likelihood given the current appearance model. Finally *Section E* presents the main contribution of this work; the multi-bag appearance model. This section describes how the use of multiple bags of subspace models can be utilised to revert to a more representative appearance model in the event that the primary model becomes ineffective, significantly increasing tracker performance. Figure 1 shows the overall flow of the algorithm and indicates which section of the paper details each of the stages.

A. Particle Filter Framework

The base of this algorithm is the particle filter [40] which allows for the estimation of the posterior probability density function of state variables characterising a dynamic system using Bayesian sequential sampling. If $\boldsymbol{Z}_{1:t-1} = \{z_1, z_2, ..., z_{t-1}\}$ denotes all previous observations up until time $t-1$ and $\boldsymbol{X}_t$ is the affine motion parameters, (t_x, t_y, w, h) where t_x and t_y describe the translation and w and h describe the width and height of a given affine parameter $\boldsymbol{X}$ in frame t. The predicted distribution of $\boldsymbol{X}_t$ is then given by $p(\boldsymbol{X}_t|\boldsymbol{Z}_{1:t-1})$ which can be recursively calculated as:

$$p(\boldsymbol{X}_t|\boldsymbol{Z}_{1:t-1}) = \int p(\boldsymbol{X}_t|\boldsymbol{X}_{t-1})\, p(\boldsymbol{X}_{t-1}|\boldsymbol{Z}_{1:t-1})\, d\boldsymbol{X}_{t-1} \tag{1}$$

Using the observation likelihood $p(\boldsymbol{Z}_t|\boldsymbol{X}_t)$ and observation $\boldsymbol{Z}_t$ and can update the state vector using Bayes rule as follows:

$$p(\boldsymbol{X}_t|\boldsymbol{Z}_{1:t}) = \frac{p(\boldsymbol{Z}_t|\boldsymbol{X}_t)\, p(\boldsymbol{X}_t|\boldsymbol{Z}_{1:t-1})}{p(\boldsymbol{Z}_t|\boldsymbol{Z}_{1:t-1})} \tag{2}$$

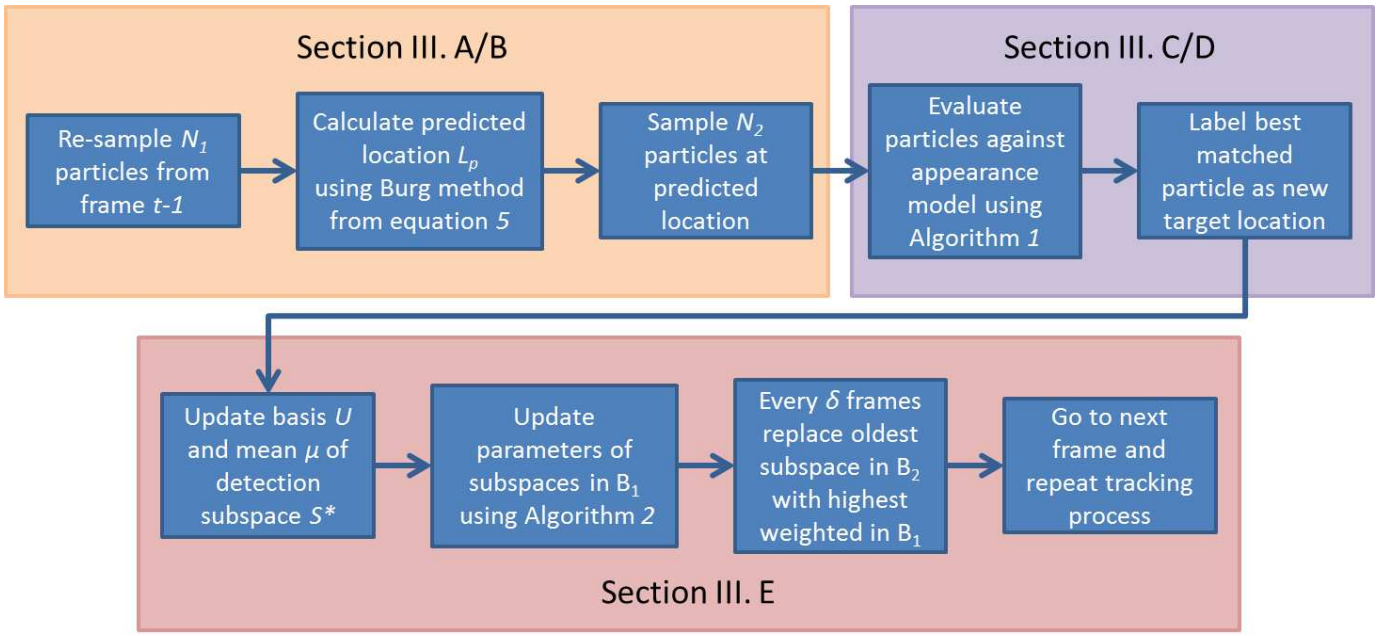


Fig. 1. Flowchart showing the general operation of the algorithm on frame t. The algorithm has been divided into three sections; the SSIR particle filter discussed in Section III A and B, the appearance model and likelihood calculation from Section III C and D and the update of the bags of subspaces presented in Section III E.

We approximate $p\left(\boldsymbol{X}_t|\boldsymbol{Z}_{1:t}\right)$, the posterior, with a set of N, $\left\{\boldsymbol{X}_t^i\right\}_{i=1:N}$ samples distributed in a Gaussian fashion, each of which are assigned a weight w_t^i which indicates its importance and is updated as:

$$w_t^i = w_{t-1}^i \frac{p\left(\boldsymbol{Z}_t|\boldsymbol{X}_t^i\right) p\left(\boldsymbol{X}_t^i|\boldsymbol{X}_{1:t-1}^i\right)}{q\left(\boldsymbol{X}_t|\boldsymbol{X}_{1:t-1}, \boldsymbol{Z}_{1:t}\right)} \tag{3}$$

where $q\left(\boldsymbol{X}_t|\boldsymbol{X}_{1:t-1}, \boldsymbol{Z}_{1:t}\right)$ is the importance distribution from which the samples $\boldsymbol{X}_t^i$ are taken and $p\left(\boldsymbol{Z}_t|\boldsymbol{X}_t^i\right)$ is the likelihood of observation $\boldsymbol{Z}_t$ given the affine parameters $\boldsymbol{X}_t^i$, the calculation of which is discussed in Section D.

The sampling importance re-sampling (SIR) filter has characteristics that allow the importance distribution to be given by $q\left(\boldsymbol{X}_t|\boldsymbol{X}_{1:t-1}, \boldsymbol{Z}_{1:t}\right) = p\left(\boldsymbol{X}_t|\boldsymbol{X}_{t-1}\right)$ and the likelihood, $p\left(\boldsymbol{Z}_t|\boldsymbol{X}_t\right)$, of the observation can be normalised to provide the weights.

This method, although generally effective, can encounter difficulties in complex situations where the weights are not accurate enough to allow for a dense sampling around the new target location. Obviously this is not optimal which is why we utilise a two stage (SSIR) particle filter which is detailed in the following section. The SSIR has the advantage of being selective in where it draws its samples by taking into account the motion of the target independently from the particle weights as well as the importance of the samples in the previous frame.

B. Particle Filter with Motion Estimation

The sampling technique utilised in our algorithm is an SSIR particle filter [38]. This is a modification of the standard SIR particle filter in that it attempts to maintain a good distribution of particles around the probable target location even in the event of an imperfect appearance model. In the SIR particle filter, the distribution of particles at time t is dependant on the particle weights at time $t-1$. In complex situations the appearance model may not be robust enough to provide weights which are accurate enough to result in a dense distribution of particles around the true target location. The SSIR aims to address this by utilising a secondary motion estimation technique to predict the new target location independent of the particle weights. As it is unlikely that the position of a target will change drastically from one frame to the next, the motion in a small number of previous frames can be used to predict the new location. Assuming that the particles with the lowest weights are the particles most likely to have been misdirected and therefore least likely to be located near the target it makes sense to attempt to improve the positioning of these particles. To this end the SSIR takes a selection of the lowest weighted particles and redistributes them around the new predicted target location in an attempt to ensure that the new target location is sampled as densely as possible.

A standard SIR particle filter draws particles from the area surrounding the previous target location based on the weights of the particles in the previous frame. We utilise this standard technique as stage one of our particle filter but acknowledge that this solution is not optimal in the case of a moving target. Rather than draw particles exclusively from the area around the previous target location, we utilise a motion estimation algorithm to direct the second stage of our particle filter to the most probable target location based on its previous trajectory before drawing samples. On a given frame t, we take the N samples from the previous frame, $t-1$, and select the N_1 particles with the largest weights which are then re-sampled to give us our first N_1 samples in the current frame. To generate our remaining N_2 samples, where $N = N_1 + N_2$ and $N_2 << N_1$, we first have to determine the predicted state of the target

based on its trajectory in the previous b frames. As the number of particles taken from N to form N_2 is very small ($N_2 << N_1$) and these particles are the N_2 particles in N with the lowest weight it is assumes that this has a minimal impact on the distribution of the remaining N_1 particles. This is because it is assumed that the particles with the lowest weights are outliers and therefore contribute little to the sampling in the following frames. We direct only a small set of particles to the estimated location to maintain as much of the standard particle filter operation as possible while increasing the probability of an accurate sampling through better utilisation of the lowest weighted particles.

To estimate the target location based on its previous trajectory we utilise a linear autoregressive filter which estimates the current value, v_t as:

$$v_t = \sum_{i=1}^{n} c_i v_{t-i} + \epsilon_t \tag{4}$$

in which ϵ is assumed to be Gaussian noise, c_i are the filter coefficients and n is the filter order. We calculate the coefficients, c_i, using the Burg algorithm [39] which aims to minimise both forward and backward prediction errors while ensuring that the Levinson-Durbin recursion is satisfied. This motion of a given target is assumed to be independent in the horizontal and vertical directions and thus we find our predicted location coordinate as:

$$L_p = (x_p, y_p) = \sum_{i=1}^{b} \left(c_i^x \boldsymbol{x}_{t-i}, c_i^y \boldsymbol{y}_{t-i}\right) \tag{5}$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are vectors containing the previous b target coordinates.

Using this predicted location, L_p, as the origin for our second particle filter and by using a Gaussian distribution to model $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ we draw the remaining N_2 particles. We then select our final N particles which are used to calculate our target likelihood as $N = N_1 \cup N_2$.

C. Appearance Model

To generate our appearance model we calculate a low-dimensional subspace generated and sequentially updated using the technique made popular by the IVT algorithm [16]. This is an incremental principal component analysis technique based on the the Sequential Karhunen-Loeve algorithm by Levy and Lindenbaum [41]. The subspace will be described by a basis consisting of a set of principal eigenvectors, $\boldsymbol{U}_i$, and the pixel-by-pixel mean of the input images, $\boldsymbol{\mu}_i$. The benefit of this approach is that, after the initial basis has been created, new data in the form of more recent images of the target object can be incorporated into the basis without the need for a complete recalculation. This means that, unlike traditional methods [42], [43], it is not necessary to store all of the image patches containing the target when tracking which quickly requires large amounts of storage. Updating the basis in this way allows for adaptation to changes in the target over time but logically if the target is changing then its new appearance must start to differ from its original appearance. To accommodate this, the update incorporates a forgetting factor. This forgetting factor us used to gradually remove old data from the basis as the new data is incorporated. The result is an appearance model which reflects the appearance of the target in a recent temporal window.

Algorithm 1 Likelihood Calculations

Require: N_1 and N_2 particles from previous frame

$match\_found = false$

for $\{\boldsymbol{B}_1\}_{i=1}^{k_1}$ **do**
  Calculate $\|\boldsymbol{\Phi}_i\|$ from (7)
  if $(min(\|\boldsymbol{\Phi}_i\|)/2) < \sigma_i$ **then**
    $match\_found = true$
    break
  end if
end for

if $!match\_found$ **then**
  Sort $\boldsymbol{B}_2$ by descending weight
  for $\{\boldsymbol{B}_2\}_{i=1}^{k_1}$ **do**
    Calculate $\|\boldsymbol{\Phi}_i\|$ from (7)
    if $(min(\|\boldsymbol{\Phi}_i\|)/2) < \sigma_i$ **then**
      $match\_found = true$
      $\boldsymbol{B}_1^{k_1} \leftarrow \boldsymbol{B}_2^i$
      break
    end if
  end for
end if

if $!match\_found$ **then**
  Calculate new subspace $\boldsymbol{S}_n$
  Calculate $\|\boldsymbol{\Phi}_i\|$ from (7) using $\boldsymbol{S}_n$
  $\boldsymbol{B}_1^{k_1} \leftarrow \boldsymbol{S}_n$
end if

To create our initial basis we utilise 100 image patches taken from the first frame. Starting with our manually labelled ground truth bounding box at location (x_g,y_g), we take 10 random translations of this image patch with the new locations (x_n,y_n) constrained independently in the x and y directions as $x_n < x_g \pm 0.01 \cdot im_w$ and $y_n < y_g \pm 0.01 \cdot im_h$ where im_w and im_h are the width and height of the image respectively. Each of these locations is then used to generate a further 10 image patches calculated as randomly scaled and rotated transformations providing us with our 100 initial image patches $\left\{\boldsymbol{I}_1^i\right\}_{i=1}^{100}$. The maximum rotation and scale variations allowed are defined as ± 0.175 radians and $\pm 1\%$ of the original size respectively. Finally we can calculate the basis, $\boldsymbol{U}_1$ through the singular value decomposition of $\left\{\boldsymbol{I}_1^i\right\}_{i=1}^{100}$ and our mean μ is an image patch consisting of the pixel-by-pixel mean of the 100 patches. Conforming with work which utilise a similar appearance model we use only the 5 principal eigenvectors of the basis in our appearance model. This has been shown to accurately capture the appearance of a target while providing a suitably small appearance model [14], [16].

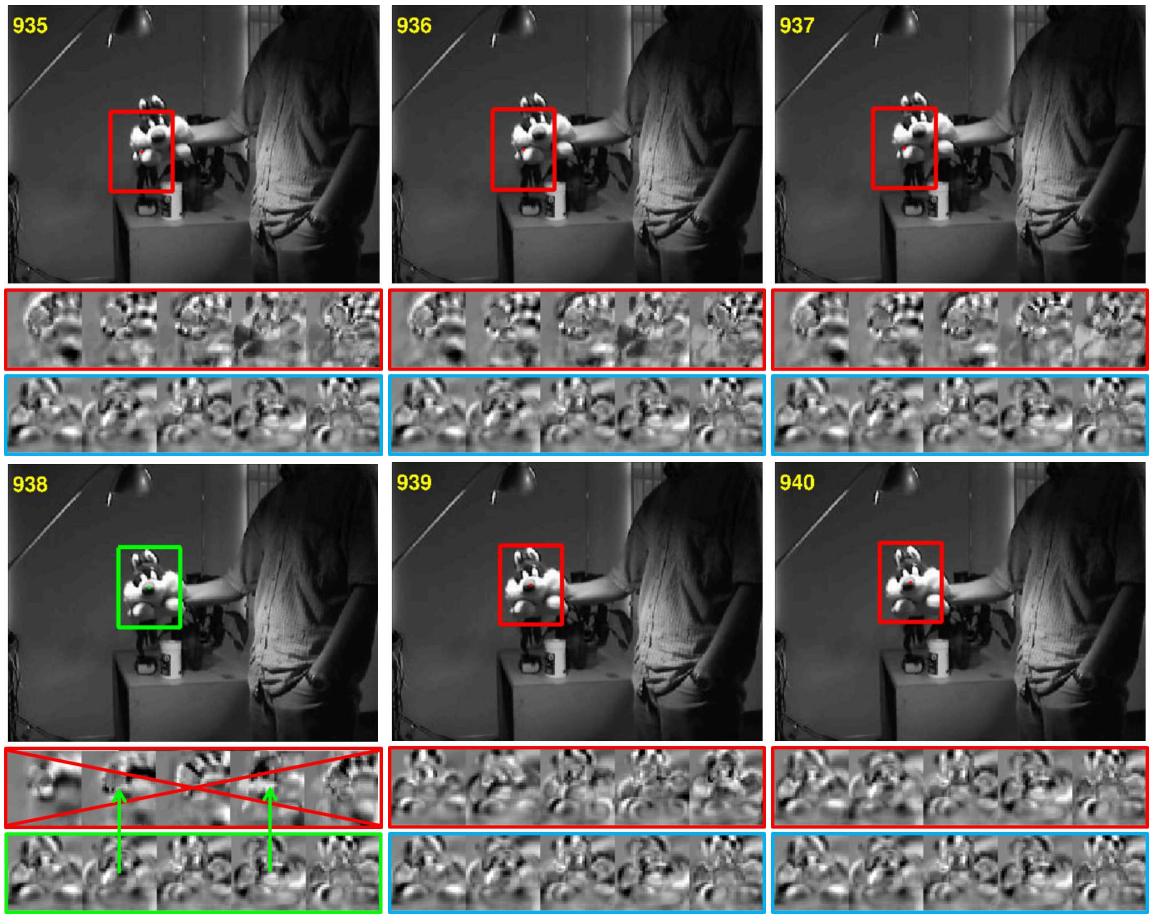


Fig. 2. 6 sequential frames of the *Sylvester* sequence showing the appearance model restoration rectifying tracker drift. For illustrative purposes we show only one subspace per bag. The subspace highlighted in red is the current tracking subspace in Bag 1 and the subspace highlighted in blue is the best subspace in Bag 2 which is restored when no subspaces in Bag 1 are sufficient.

D. Target Likelihood

In order to determine the likelihood that a given image patch $\boldsymbol{I}_t$ accurately represents the target, we evaluate the reconstruction error $\|\boldsymbol{\Phi}_i\|$ between the image patch and a subspace $\boldsymbol{S}_i$ within the appearance model. This is based on the theory that an image patch which contains the target will be well represented within the basis and can therefore be accurately reconstructed. Assuming that the likelihood is inversely proportional to the reconstruction error [24], [38] the image patch with the lowest error is most likely to accurately represent the target object.

The image patch $\boldsymbol{I}_t$ can be reconstructed from subspace $\boldsymbol{S}_i$ using equation (6), resulting in the reconstructed image patch $\hat{\boldsymbol{I}}_{t,i}$.

$$\hat{\boldsymbol{I}}_{t,i} = \boldsymbol{U}_i \boldsymbol{U}_i^T \left(\boldsymbol{I}_t - \boldsymbol{\mu}_i\right) + \boldsymbol{\mu}_i \tag{6}$$

Taking the l_2 norm of the pixel-wise difference of the original and reconstructed image patches gives the reconstruction error as follows:

$$\|\boldsymbol{\Phi}_i\| = \left\| \boldsymbol{I}_t - \hat{\boldsymbol{I}}_{t,i} \right\| \tag{7}$$

As the likelihood is inversely proportional to the reconstruction error it can be represented as:

$$p\left(\boldsymbol{I}_t | \boldsymbol{S}_i, \boldsymbol{X}_t\right) \propto exp(-\eta \left\|\boldsymbol{\Phi}_i\right\|^2) \tag{8}$$

with a constant value η. The final probability can then be expressed as:

$$p\left(\boldsymbol{I}_t | \boldsymbol{X}_t\right) \propto \omega_i \sum_i exp(-\eta \left\|\boldsymbol{\Phi}_i\right\|^2) \tag{9}$$

where ω_i is the weight factor of the i^{th} subspace, the calculation of which is discussed in the following section. This allows us to find the image patch with the largest weight $\boldsymbol{I}_t^*$ and its location parameters $\boldsymbol{X}_t^*$ by normalising the probability from (9) and selecting the maximum value. We will also refer to the subspace against which the best match was made as the detection subspace; $\boldsymbol{S}_i^*$.

E. Bags of Subspaces

The primary contribution of this work is the manner in which we store and utilise previous appearance models in multiple bags to allow for model restoration in the event that the primary models become unable to track the target. Rather than have a single appearance model which is updated temporally as in algorithms such as IVT [16] and KCF [23] we use two bags of appearance models. When updating an individual model we utilise the technique employed in [16] but attempt to remove the negative implications of an

uncensored temporal model update. To achieve this, we utilise 2 bags of subspaces, $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$ where each subspace is an individual appearance model. The bag $\boldsymbol{B}_1$ is the primary subspace bag and contains the current subspace being used in the tracking process and $k_1 - 1$ other subspaces which are initialised as copies of the primary subspace such that $\{\boldsymbol{B}_1\}_{i=1}^{k_1} = \{\boldsymbol{S}_1, \boldsymbol{S}_2, ..., \boldsymbol{S}_{k_1}\}$ where $\boldsymbol{S}_i$ denotes a single appearance model. These subspaces are updated based on their tracking success over time. The bag $\boldsymbol{B}_2$ contains a further k_2 subspaces which are temporally distributed snapshots of the current tracking subspace. These snapshots are taken every δ frames and if the bag is full (i.e. contains k_2 subspaces), the oldest subspace is replaced to maintain a temporal window of subspaces. Through the remainder of this work we will use the following notation when discussing the bags of subspaces. $\boldsymbol{B}_1$ refers to the entirety of bag 1 and all subspaces contained within it while $\boldsymbol{B}_1^2$ refers to the second subspace within $\boldsymbol{B}_1$. Notation such as $\{\boldsymbol{B}_1\}_{i=1}^{3}$ would refer to subspaces $i = 1:3$ within $\boldsymbol{B}_1$.

The concept behind this multi-bag approach is that $\boldsymbol{B}_1$ contains a selection of subspaces which have been very successful at one time during the tracking process and therefore have a high weight regardless of their age. $\boldsymbol{B}_2$ on the other hand captures the successful subspace temporally as tracking progresses independent of how long that particular subspace has been tracking or how highly it is weighted. The combination of these two bags gives the algorithm a much better chance of not just recovering from poor model updates but doing so in a way that is most beneficial to successful tracking based on the target appearance at that time. This is useful in recovering from occlusion, out of view targets and tracker drift.

To describe how these bags are utilised and how the subspaces within them are updated we will refer to algorithm 1 which shows how the likelihood of a given set of particles is evaluated against these bags. Before this discussion can take place it should be noted that each subspace has several parameters associated with it such that $\boldsymbol{S}_i = \{\boldsymbol{U}_i, \boldsymbol{\mu}_i, \omega_i, c_i, f_i, \sigma_i\}$ which represent the basis ($\boldsymbol{U}_i$), mean ($\boldsymbol{\mu}_i$), weight (ω_i), tracking count (c_i), forgetting factor (f_i) and variance of the error (σ_i).

During the tracking process, the subspaces in $\boldsymbol{B}_1$ are ordered by highest weight before the tracking process begins. We start with the highest weighted subspace and evaluate the error using (7). The minimum error, relating to the particle with the highest probability, is compared to the variance of the error associated with the subspace, σ_i, the calculation of which is detailed later in this section. The subspace is only deemed to have correctly detected the target if half the minimum error is less than σ_i, otherwise we move to the next subspace in $\boldsymbol{B}_1$ and again attempt to locate the target. Assuming that one of the subspaces in $\boldsymbol{B}_1$ returns an error which satisfies the variance test we will utilise this subspace for tracking in this frame and do not need to evaluate the remaining subspaces in $\boldsymbol{B}_1$ or any of $\boldsymbol{B}_2$. We will refer to this detection subspace as $\boldsymbol{S}^*$ throughout the remainder of this section.

Every δ frames the tracking subspace $\boldsymbol{S}^*$ is copied to $\boldsymbol{B}_2$ where it is stored for future use under the assumption that it was successful at one point in time and therefore may be useful in the future. If the entirety of $\boldsymbol{B}_1$ is evaluated and no suitable match is determined we move to $\boldsymbol{B}_2$ and attempt to revert to a previously successful appearance model to continue tracking. Each of the subspaces contained in this bag are evaluated in the same manner as $\boldsymbol{B}_1$ starting with the highest weighted and descending progressively. If one of the subspaces in $\boldsymbol{B}_2$ satisfies the variance test then the lowest weighted subspace in $\boldsymbol{B}_1$ is replaced with the matched subspace from $\boldsymbol{B}_2$ and thus the appearance model has been restored to a state more representative of the target at this time.

There is, of course, the possibility that all of the subspaces in $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$ fail. In this case we create a new subspace, $\boldsymbol{S}_n$ which is calculated from the image patch from frame 1 along with the image patches from the previous γ frames and uses all of the standard parameters given in Section IV. We then find the minimum error against this subspace and select that location as the new target and the lowest weighted subspace in $\boldsymbol{B}_1$ is replaced with $\boldsymbol{S}_n$. As this will only ever occur in the event that $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$ fail we use this as a last resort in an attempt to continue tracking when all other subspaces have become ineffective.

The final stage after the detection of the target object is to update the basis and the mean of the subspace which successfully carried out the detection ($\boldsymbol{S}^*$) and also to recalculate several of the parameters associated with each of the subspaces in $\boldsymbol{B}_1$. Updating the mean and basis is carried out using the incremental principal component analysis technique popularised by Ross *et. al* in the IVT algorithm [16]. Depending on the result of the tracking process, certain parameters associated with each subspace must be modified as shown in Algorithm 2. For each subspace in $\boldsymbol{B}_1$, the weight (ω_i) is recalculated as [38]:

$$\omega_i = (1 - \Lambda_\omega)\omega_i + \alpha\Lambda_\omega \tag{10}$$

The weight of a subspace is only ever increased if that subspace is $\boldsymbol{S}^*$, the detection subspace, in which case the value of α is set to 1. For all other subspaces, α is set to 0 and the weight is reduced due to the update parameter Λ_ω. For the detection subspace, the tracking counter c_i is incremented which allows the forgetting factor of the subspace, f_i, to be updated using [38]:

$$f_i = f_o \cdot \left(\frac{\varphi}{c_i}\right) \tag{11}$$

where f_o and φ are static parameters. This forgetting factor will only be updated in the event that the subspace is the detection subspace as only then will the value of c_i change. Finally the variance of the error, σ_i, associated with the detection subspace is updated with respect to the update parameter Λ_v as follows [38]:

$$\sigma_i = \sqrt{(1 - \Lambda_v)\,\sigma_i^2 + \Lambda_v \left\|\boldsymbol{\Phi}_i\right\|^2} \tag{12}$$

Figure 2, which shows 6 consecutive frames of the *Sylvester* sequence, illustrates the multi-bag operation more clearly. For illustrative purposes we show only one subspace in $\boldsymbol{B}_1$, indicated in red, and one subspace in $\boldsymbol{B}_2$ indicated in blue. The figure shows that the current tracking subspace (red) in frames 935-937 has started to drift due to error accumulation.

Algorithm 2 Subspace Parameter Recalculation

Require: Bag $\boldsymbol{B}_1$
for $\{\boldsymbol{B}_1\}_{i=1}^{k_1}$ **do**
  $\alpha = 0$
  if $\boldsymbol{S}_i$ is the matched subspace **then**
    $\alpha = 1$
    Increment c_i
    Update error variance (σ_i) using (12)
    Update forgetting factor (f_i) using (11)
  end if
  Update subspace weight (ω_i) using (10)
end for

In frame 938 the error becomes too high and the subspace fails to pass the variance test. At this time, as all other subspaces in $\boldsymbol{B}_1$ also fail to track, the best subspace from $\boldsymbol{B}_2$ is moved to $\boldsymbol{B}_1$ and is used to correct the drift and continue tracking as indicated in green. This appearance model restoration significantly increases the tracker accuracy and its robustness to drift and occlusion. It should be noted that in the event that this second bag of subspaces is never utilised, the algorithm should theoretically exhibit performance similar to the SSIR algorithm [38] and this is discussed further in Section V.

IV. Experimental Procedure and Results

The evaluation of our algorithm was carried out on a range of challenging publicly available image sequences from [44] and we compare our results against 14 of the state of the art tracking algorithms; ASLA [45], CSK [22], CT [26], DFT [30], DSST [46], FCT [47], IVT [16], KCF(grayscale features) [23], KCF_HOG(HOG features) [23], L1 [48], MIL [36], Struck [49] and TLD [11]. We also include our implementation of SSIR [38] as code is not currently available.

In the interest of fair comparison of algorithms, we use static parameters throughout all testing so that no algorithms are tuned to specifically meet the requirements of a given image sequence. For consistency, if an algorithm has parameters which can be tuned, we use the parameters set by Wu *et al.* [44] when they carry out their algorithm evaluation or the parameters set by default in the code provided by the authors.

In our algorithm, the following parameters were used and remained constant throughout the testing procedure. For the particle filter we set $N = 600$ as the total number of particles, $N_1 = 500$ particles from the previous frame, $N_2 = 100$ particles from the autoregressive prediction and the diagonal covariance matrix with elements that define the variance of the affine transformation parameters is fixed as $\psi = \{\sigma_x^2, \sigma_y^2, \sigma_s^2\} = \{9^2, 9^2, 6^2\}$ corresponding to the x and y translation and the scale respectively [24].

In the autoregressive Burg calculations, we take into account the object trajectory over the past $b = 6$ frames and set $n = 2$ for a second order filter [24]. Each time a subspace is created, it is initialised with the following parameters; $\omega_i = 0.1$, $c_i = 0$, $f_i = 0.5$ and $\sigma_i = 5$. The update parameters Λ_ω, Λ_v, f_o and φ are set to 0.01, 0.02, 0.99 and 0.5 respectively [38].

When a subspace is moved into $\boldsymbol{B}_2$, it retains its current weight, forgetting factor and error variance but its tracking count c_i is reset to 0. The lengths of $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$ were empirically set to $k_1 = k_2 = 3$ respectively. The use of bag sizes greater than this seemed to have little benefit to the tracking performance of the algorithm but obviously had greater computational overheads. The highest weighted subspace from $\boldsymbol{B}_1$ is copied to $\boldsymbol{B}_2$ every $\delta = 10$ frames. In the event that a new subspace must be created, we use the previous $\gamma = 5$ frames along with frame 1, the introduction of which has been shown to reduce tracker drift [29].

Our performance evaluation is carried out using 2 independent metrics, centre location error (CLE) and percentage area overlap (Score) both of which are commonly used in the evaluation of object trackers [50]. CLE is calculated as the distance in pixels between the centre of the ground truth bounding box (G) and the tracker bounding box (T):

$$CLE = \sqrt{(G_x - T_x)^2 + (G_y - T_y)^2} \tag{13}$$

while the Score is calculated as:

$$Score = \frac{area\,(G \cap T)}{area\,(G \cup T)} \tag{14}$$

Tables I and II show the average CLE and average Score for all of the evaluated trackers across the range of 20 image sequences. It should be noted that the TLD algorithm often fails to report a bounding box for frames in a sequence making any CLE calculations inaccurate and as a result no CLE is reported for the TLD on these sequences [51]–[53]. The graphs in Figure 3 show the CLE for the 6 trackers with the lowest average CLE (OURS, KCF_HOG, SSIR, STRUCK, KCF and DSST). For clearer visualisation we do not show the full error range and instead focus on a smaller range. While this can result in some of the higher errors being missing from the graphs it allows for a clearer representation of the lower errors where the algorithms are actually tracking the target.

The operating speeds of the algorithms compared in this paper vary greatly from 332 frames per second (KCF) to 2 frames per second (ASLA). Despite the use of multiple bags of subspaces, our algorithm, which operates at an average of 9 frames per second has a speed comparable to that of many popular algorithms such as ASLA, DFT, IVT, L1 and MIL which operate at an average of 2, 13, 16, 13 and 5 frames per second respectively. In terms of computational complexity we will relate our algorithm to the complexity of the IVT algorithm which also utilises a particle filter but with a single subspace appearance model. The computational complexity of the IVT algorithm increases linearly with the number of particles utilised giving a complexity of $O(N)$ when using N particles. Our algorithm mirrors this complexity for the majority of its operation, i.e. when the first subspace in B_1 is used for tracking. At the other end of the scale, the complexity of our algorithm can increase to $O(N * (k_1 + k_2 + 1))$. This is because we have to evaluate the N particles k_1 times to determine that none of the subspaces in B_1 are effective, a further k_2 times to determine that the subspaces in B_2 are not effective. In this worst case scenario we then create a new subspace which must be evaluated against the N particles.

TABLE I
AVERAGE CENTRE LOCATION ERROR (CLE) FOR THE ALGORITHMS ON THE SELECTED IMAGE SEQUENCES WHERE **BOLD RED** INDICATES THE BEST RESULT AND *Italic Blue* INDICATES SECOND BEST.

Sequence	ASLA	CSK	CT	DFT	DSST	FCT	IVT	KCF	KCF_HOG	L1	MIL	STRUCK	TLD	SSIR	OURS
Basketball	125.39	**6.53**	122.08	18.03	10.80	92.42	116.54	126.94	*7.89*	150.25	103.80	215.72	–	51.53	54.49
Car4	1.93	19.30	81.10	62.25	**1.73**	77.14	*1.90*	36.21	10.11	87.45	67.28	9.01	–	12.93	8.66
CarDark	*0.99*	3.60	120.36	58.99	1.82	47.28	8.23	4.79	6.30	17.89	44.60	**0.91**	–	2.83	2.83
Couple	107.40	144.73	35.59	108.78	125.76	37.22	108.85	*17.73*	47.78	98.27	38.40	**10.15**	–	77.94	38.20
David2	1.64	2.57	79.56	17.38	2.15	14.33	**1.43**	5.69	2.13	1.86	75.74	*1.47*	4.54	3.01	3.01
Deer	140.90	*4.83*	235.56	98.45	16.65	9.12	194.30	**4.51**	21.14	111.10	217.12	5.19	–	10.22	10.22
Dog1	5.59	3.78	10.29	41.03	4.42	8.89	**3.51**	*3.53*	4.38	4.25	16.43	5.54	13.94	6.82	7.28
Dudek	13.30	19.22	19.83	18.74	13.47	33.53	*10.66*	15.67	11.33	37.70	150.92	11.57	–	10.71	**10.36**
FaceOcc2	7.86	*5.91*	10.28	7.78	6.84	15.50	8.00	**5.55**	7.75	14.34	17.57	6.18	–	8.22	10.15
Fish	3.97	41.35	25.83	8.75	4.12	11.88	5.09	7.69	*3.84*	74.10	12.67	**3.40**	–	6.48	6.47
Football1	10.80	16.38	11.03	**1.83**	9.23	23.28	23.85	18.11	5.40	25.01	13.53	26.90	6.17	*4.85*	*4.85*
Girl	20.82	19.36	18.78	23.86	11.03	15.90	23.91	25.11	11.95	**3.36**	18.86	18.41	–	*10.38*	*10.38*
Jumping	47.31	86.08	46.35	67.39	36.93	37.39	61.34	32.11	26.33	54.33	12.78	*6.77*	–	**6.27**	**6.27**
Lemming	185.48	114.21	82.97	77.76	81.93	68.54	184.00	58.25	77.84	169.84	69.85	*37.64*	–	13.99	**11.64**
MHYang	*2.25*	3.76	24.15	8.93	2.30	14.89	**1.81**	3.50	3.74	3.05	32.16	2.68	–	4.96	3.39
MotorRolling	195.80	434.11	*162.83*	174.03	296.67	165.76	169.64	188.47	228.42	203.82	166.83	**149.27**	–	183.91	205.48
MountainBike	**5.60**	*6.52*	210.73	154.82	7.84	11.71	7.36	6.86	7.55	7.32	216.37	8.63	–	10.01	13.62
Singer1	*3.62*	182.42	18.26	18.70	**3.30**	19.10	11.75	18.47	12.85	5.14	19.34	14.09	22.96	14.17	13.57
Sylvester	8.10	9.93	13.54	44.85	13.53	*7.82*	36.75	13.77	12.75	19.84	44.77	**6.02**	14.79	35.65	11.12
Tiger1	50.47	73.08	36.28	*9.61*	18.35	19.70	99.56	44.89	**8.04**	98.43	103.15	50.18	–	70.76	42.51
Average	46.96	59.88	68.27	51.10	33.44	36.57	53.92	31.89	*25.88*	59.37	72.11	29.49	–	27.28	**23.72**

TABLE II
AVERAGE AREA OVERLAP (SCORE) FOR THE ALGORITHMS ON THE SELECTED IMAGE SEQUENCES WHERE **BOLD RED** INDICATES THE BEST RESULT AND *Italic Blue* INDICATES SECOND BEST.

Sequence	ASLA	CSK	CT	DFT	DSST	FCT	IVT	KCF	KCF_HOG	L1	MIL	STRUCK	TLD	SSIR	OURS
Basketball	28.66	**71.32**	20.87	61.36	58.42	23.21	14.75	22.71	*68.75*	1.94	21.61	2.46	12.81	24.75	24.04
Car4	79.78	46.87	23.96	23.97	**89.92**	23.95	*87.42*	35.69	48.69	15.38	23.06	49.41	0.88	58.86	65.41
CarDark	78.02	72.34	0.31	38.41	*81.19*	14.55	67.18	66.26	61.72	50.62	0.48	**89.57**	29.77	73.17	73.17
Couple	8.89	7.49	48.10	7.76	9.01	*48.45*	7.13	47.63	20.07	19.59	43.41	**55.88**	36.60	22.36	37.06
David2	*87.23*	82.96	0.27	54.69	82.77	45.14	66.13	65.41	84.34	77.38	0.93	**88.61**	70.51	71.04	71.04
Deer	3.88	*75.04*	3.96	25.67	64.65	68.23	3.21	**75.22**	62.45	9.99	7.98	74.22	5.55	70.26	70.26
Dog1	69.06	55.15	52.22	43.87	*76.23*	47.73	73.98	55.74	55.54	**81.64**	45.23	55.17	56.46	69.77	69.27
Dudek	*76.79*	48.44	69.70	69.16	**79.16**	59.18	75.49	61.54	60.67	52.61	28.45	72.82	50.00	74.77	74.31
FaceOcc2	76.63	78.20	70.93	77.05	*78.40*	65.43	67.05	78.37	75.35	25.25	60.85	**78.56**	55.17	73.72	72.27
Fish	83.93	21.48	43.00	76.01	80.16	66.81	77.52	77.65	*84.14*	9.99	66.84	**86.12**	28.23	68.42	67.64
Football1	52.21	46.22	45.68	**87.07**	50.66	18.01	55.32	45.80	*70.38*	28.41	38.16	36.30	60.51	65.05	65.05
Girl	22.31	37.17	28.32	29.11	44.94	35.75	17.25	52.37	*54.87*	**57.32**	27.33	37.18	36.75	39.27	39.27
Jumping	7.59	5.02	5.92	11.09	14.49	20.78	12.33	17.30	28.00	7.85	40.99	**60.95**	7.38	*58.45*	*58.45*
Lemming	14.18	33.32	31.54	40.75	33.07	**49.94**	12.43	38.58	38.75	12.67	40.05	*48.40*	29.14	43.02	43.59
MHYang	**90.35**	79.84	43.86	71.18	80.82	59.85	77.61	79.82	79.91	78.22	34.21	*81.81*	57.83	70.33	71.46
MotorRolling	10.38	0.29	10.79	8.52	9.84	12.91	8.95	10.68	10.28	8.37	6.87	**16.91**	8.85	13.06	*13.17*
MountainBike	**77.72**	71.98	14.80	29.89	73.13	65.21	*74.28*	70.64	71.82	64.09	12.10	70.54	30.39	68.71	62.05
Singer1	*82.67*	0.09	35.20	35.78	**83.41**	35.85	56.70	27.69	29.88	74.27	32.37	36.58	48.70	49.09	48.81
Sylvester	67.83	63.01	62.52	38.02	63.22	*68.06*	51.83	63.57	65.37	35.09	24.90	**72.99**	54.87	44.89	65.86
Tiger1	33.85	26.94	38.04	*75.06*	61.69	57.76	12.23	46.04	**78.78**	14.69	9.44	41.19	6.55	21.84	36.19
Average	52.60	46.16	32.50	45.22	**60.76**	44.34	45.94	51.93	*57.49*	36.27	28.26	57.78	34.35	54.04	56.42

The algorithm complexity will only realistically be higher than $O(N)$ for occasional individual frames during the tracking process.

V. DISCUSSION OF RESULTS

In this section we present an analysis of the results in the previous section as well as an in depth discussion of 6 of these sequences which present particularly challenging features. Figures 4, 5 and 6 show a selection of frames from each of these sequences including the output of the 6 algorithms with the lowest average CLE (OURS, KCF_HOG, SSIR, STRUCK, KCF and DSST). We display only the top 5 algorithms as this makes clear visualisation much easier and is of more benefit to the analysis of the performance of our algorithm.

We begin with an analysis of Tables I and II which compare the average CLE and Score of the 15 algorithms on 20 benchmark sequences. It is clear from these tables that our algorithm produces results which can be compared directly

TABLE III
A COMPARISON OF THE 16 SEQUENCES WHERE OUR ALGORITHM UTILISES $\boldsymbol{B}_2$ SHOWING AN OVERALL IMPROVEMENT OF AVERAGE CENTRE LOCATION ERROR.

Sequence	SSIR	OURS
Basketball	51.53	54.49
Car4	12.93	8.66
Couple	77.94	38.20
Dog1	6.82	7.28
Dudek	10.71	10.36
FaceOcc2	8.222	10.15
Fish	6.48	6.47
Lemming	13.99	11.64
MHYang	4.96	3.39
MotorRolling	183.91	205.48
MountainBike	10.01	13.62
Singer1	14.17	13.57
Sylvester	35.65	11.12
Tiger1	70.76	42.51
Average	36.29	31.21

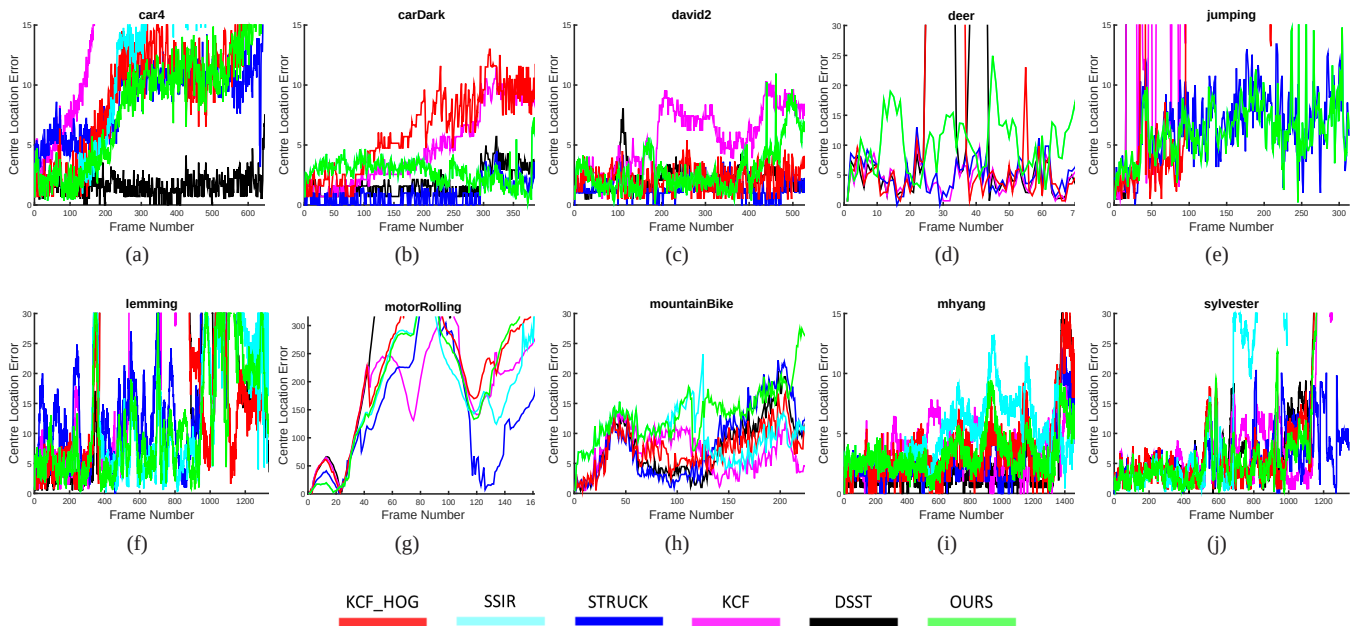


Fig. 3. Plots of the centre location error of the algorithms on each of the image sequences.

Fig. 4. Visual comparison of algorithm performance on 2 challenging sequences; *Car4* and *CarDark*.

TABLE IV
A COMPARISON OF THE 16 SEQUENCES WHERE OUR ALGORITHM UTILISES B_2 SHOWING AN OVERALL IMPROVEMENT OF AVERAGE AREA OVERLAP SCORE.

Sequence	SSIR	OURS
Basketball	24.75	24.04
Car4	58.86	65.41
Couple	22.36	37.06
Dog1	69.77	69.27
Dudek	74.77	74.31
FaceOcc2	73.72	72.27
Fish	68.42	67.64
Lemming	43.02	43.59
MHYang	70.33	71.46
MotorRolling	13.06	13.17
MountainBike	68.71	62.05
Singer1	49.09	48.81
Sylvester	44.89	65.86
Tiger1	21.84	36.19
Average	50.26	53.65

with a selection of the state of the art algorithms. The results produced show a performance which is often higher than several of the presented trackers and in some instances our algorithm produces the best result. It should also be noted that our algorithm achieves the best average result in the CLE evaluation. The CLE graphs in Figure 3 support this and show that in the majority of cases our algorithm exhibits similar trends to the other top 5 trackers presented. At present the algorithm does not achieve as strong results in the Area Overlap category as the scale adaptation is limited. This results in other scale adaptive algorithms such as STRUCK and DSST achieving better results in this case.

Directly comparing our results to that of the SSIR it can be seen that on 6 of the 20 sequences (*CarDark*, *David2*, *Deer*, *Football1*, *Girl* and *Jumping*) the two algorithms produce the same result which indicates that our B2 was never utilised and that B1 was maintained well enough to track the target successfully. This is confirmed by the results which show that,

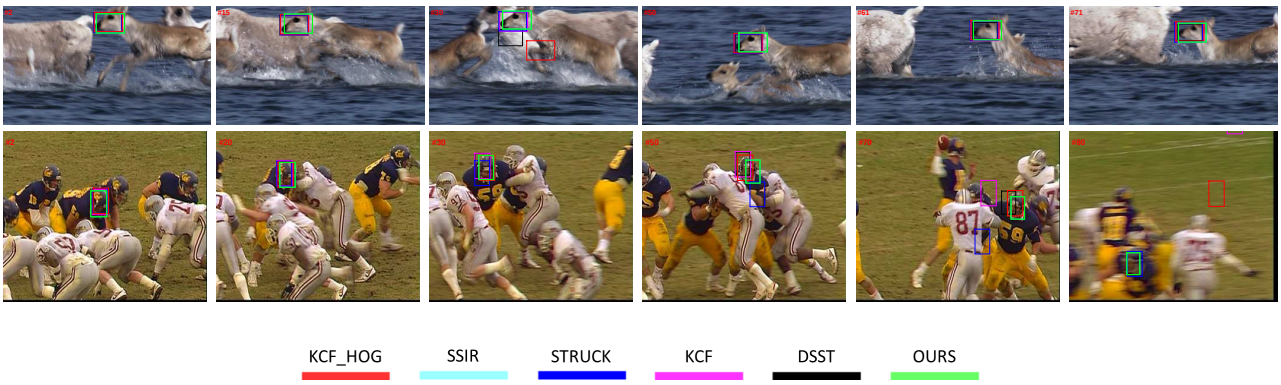


Fig. 5. Visual comparison of algorithm performance on 2 challenging sequences; *Deer* and *Football1*.

while the algorithm does not always achieve the best result, the Area Overlap and CLE indicate successful tracking. Analysis of the remaining 14 results where B2 does activate shows an overall improvement of the average CLE and Score of the SSIR algorithm. This is highlighted in Tables III and IV which directly compare these two algorithms.

Moving now to a discussion of the 6 image sequences depicted in Figures 4 - 6 which show the performance of the top 6 algorithms at certain points throughout the sequences:

Car4: The main challenges in this image sequence (Figure 4) are scale change and illumination variation, both of which are significant throughout. The scale change, while gradual, continues throughout the majority of the sequence which presents a problem for many algorithms. This is due to the possibility of learning background information in the event that the bounding box is not accurately resized. Between frames 181 and 290 the vehicle passes under a bridge which causes a significant step change in the illumination of the target. While this presents only a slight visual change from human perception it results in a drastic step change in the pixel values. Our algorithm is unaffected by these illumination changes as the subspace appearance model has some inherent robustness to such variations. As a result the algorithm continues to track through both the step decrease and increase of illumination caused by the bridge. The element of this sequence which causes our algorithm to struggle is the scale change. It can be seen that while some degree of resizing of the bounding box occurs to accommodate the change in scale it is not completely effective in this instance.

CarDark:The most challenging issue in this sequence (Figure 4) is the repeated significant variation in illumination caused by oncoming vehicles. There is also the issue that the sequence is very dark to begin with and is fairly low resolution. Having such a small target means that there is a lot less information available to learn in the appearance model. All of the top algorithms track this sequence but it is a good indication of our algorithms robustness to low resolution and illumination variation as well as the ability to track a target which is similar in appearance to many of the other objects in the scene.

Deer: Sometimes referred to as *Animal* in the literature, the target in this sequence (Figure 5) is extremely fast moving. This means that traditional sampling techniques often fail as the step distance covered by the target on a frame to frame basis can put it outwith the designated search window of the algorithm. This is illustrated in frame 30 where the DSST and the KCF_HOG algorithms do not manage to track consistently. This is reflected in the graph in Figure 3. Our particle filter approach in this instance proves to be robust enough to handle this fast motion but the generation of the appearance model from the first frame also contributes to the success. The IVT algorithm also uses a particle filter but only generates a subspace appearance model after 5 frames by which time it has already failed. This robustness to fast motion is a very important attribute in object tracking.

Football: This sequence (Figure 5), while being quite short, contains significant motion blur, occlusion, in-plane and out-of-plane rotation. There is also noticeable background clutter as there are many similar targets crowded in the scene. It is clear that throughout the sequence many of the algorithms become lost and are unable to track the target to the end. Our algorithm manages to remain accurate throughout the entire sequence.

Jumping: Motion blur is the primary cause of difficulty in this sequence (Figure 6). The fast movement of the target in combination with camera motion causes the target to blur significantly throughout the sequence. Among the top performing algorithms the KCF, KCF_HOG and the DSST algorithms show difficulty in following the target under these conditions. The appearance model and sampling method employed by our algorithm proves robust enough to track the target smoothly throughout the sequence. Static cameras are not always available in tracking applications and a sufficient frame-rate to avoid blur cannot always be guaranteed. For this reason the ability to track targets under these conditions is important.

Lemming: This sequence is possibly one of the most challenging in the benchmark dataset (Figure 6). Containing scale change, occlusion , out of view target, significant pose variation, fast motion, motion blur and severe background clutter; this target is quickly lost by the vast majority of algorithms available. At frame 340 the target becomes fully

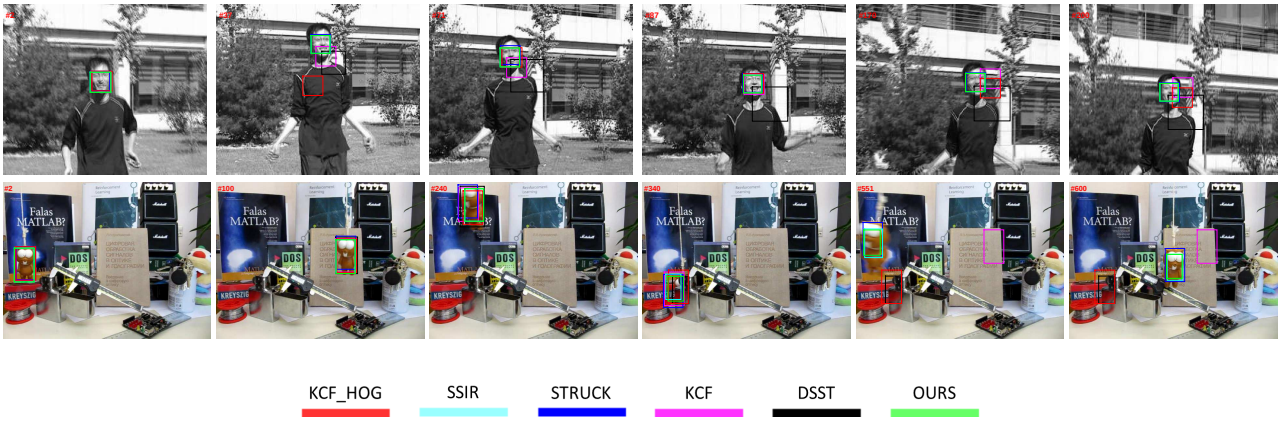


Fig. 6. Visual comparison of algorithm performance on 2 challenging sequences; *Jumping* and *Lemming*.

occluded and remains in this state for many frames. This gives some incremental update algorithms like the DSST and KCF_HOG time to learn the appearance of the background and as a result, fail to relocate the target when it emerges. At this time, our model restoration system activates and we revert to an appearance model which was stored before the occlusion allowing us to re-detect the target and continue tracking throughout the remainder of the sequence with minimal error. It is situations like this that show the benefits of our multi-bag subspace appearance model. As with the *Car4* sequence it is the change in scale that our algorithm struggles with and while we achieve a good CLE the scale adjustment means that out Score is not as high as STRUCK which has the second best CLE but seems to handle scale changes more easily.

The sequences discussed above contain all of the characteristics of object tracking sequences that are seen as problematic for tracking algorithms to deal with. These challenges include; Illumination Variation, Scale Change, Low Resolution, Fast Motion, Motion Blur, Background Clutter, Occlusion In-Plane and Out-of-Plane Rotation. Our results indicate that our algorithm is capable of dealing with most of these challenges and produces results which compete with the selection of state of the art algorithms presented. The challenge which causes the most difficulty for our algorithm is scale change. This often results in our algorithm achieving a good CLE but a slightly lower Score, often being beaten by STRUCK which deals with scale changes very well.

VI. Conclusion

This paper presented a novel object tracking method which produces very promising results in comparison to several state of the art, publicly available tracking algorithms. The algorithm utilises a two stage SSIR particle filter which takes into account object trajectory using an autoregressive filter technique. This, in combination with an incrementally updated subspace based appearance model and a reconstruction error likelihood function forms the foundation of our algorithm. The primary contribution of this work is the appearance model restoration technique which utilises two bags of subspaces. The first bag contains the primary tracking models while the second bag contains a set of temporally buffered models which the algorithm can choose to revert to in the case that the main appearance models fail. This has proven highly effective in recovering from tracker drift as well as partial and full occlusion and has resulted in a accurate, robust algorithm which is able to produce competitive results in comparison to a range of trackers.

One future aim with this work is to incorporate adaptive bag sizes to increase robustness to long term occlusion. We also plan increase the robustness to scale changes and implement a re-detection scheme [11], [54] to locate the object in the event that it leaves and re-enters the frame or moves significantly while occluded and is therefore outside the search radius of the particle filter. There is also the possibility of introducing adaptive update parameters for the subspace weights and error variance to reflect the confidence of the subspace rather than the current constant increase or decrease method implemented at present.

References

[1] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM computing surveys (CSUR)*, vol. 38, no. 4, p. 13, 2006.
[2] K. Cannons, "A review of visual tracking," *Dept. Comput. Sci. Eng., York Univ., Toronto, Canada, Tech. Rep. CSE-2008-07*, 2008.
[3] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song, "Recent advances and trends in visual tracking: A review," *Neurocomputing*, vol. 74, no. 18, pp. 3823–3831, 2011.
[4] B. Benfold and I. Reid, "Stable multi-target tracking in real-time surveillance video," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 3457–3464, IEEE, 2011.
[5] G. R. Bradski, "Real time face and object tracking as a component of a perceptual user interface," in *Applications of Computer Vision, 1998. WACV'98. Proceedings., Fourth IEEE Workshop on*, pp. 214–219, IEEE, 1998.
[6] N. Bellotto and H. Hu, "Multisensor-based human detection and tracking for mobile service robots," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 39, no. 1, pp. 167–181, 2009.
[7] A. Vu, A. Ramanandan, A. Chen, J. A. Farrell, and M. Barth, "Real-time computer vision/dgps-aided inertial navigation system for lane-level vehicle navigation," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, no. 2, pp. 899–913, 2012.
[8] M. Cristani, R. Raghavendra, A. Del Bue, and V. Murino, "Human behavior analysis in video surveillance: A social signal processing perspective," *Neurocomputing*, vol. 100, pp. 86–97, 2013.
[9] F. Yang, H. Lu, and M.-H. Yang, "Robust superpixel tracking," *Image Processing, IEEE Transactions on*, vol. 23, no. 4, pp. 1639–1651, 2014.

[10] S. Avidan, "Support vector tracking," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 8, pp. 1064–1072, 2004.

[11] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 7, pp. 1409–1422, 2012.

[12] D. Du, L. Zhang, H. Lu, X. Mei, and X. Li, "Discriminative hash tracking with group sparsity," *Cybernetics, IEEE Transactions on*, in press, 2015.

[13] D. Wang, H. Lu, and M.-H. Yang, "Least soft-threshold squares tracking," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pp. 2371–2378, IEEE, 2013.

[14] D. Wang, H. Lu, and M.-H. Yang, "Online object tracking with sparse prototypes," *Image Processing, IEEE Transactions on*, vol. 22, no. 1, pp. 314–325, 2013.

[15] X. Mei and H. Ling, "Robust visual tracking using l1 minimization," in *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 1436–1443, IEEE, 2009.

[16] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 125–141, 2008.

[17] Z. H. Khan and I. Y.-H. Gu, "Nonlinear dynamic model for visual object tracking on grassmann manifolds with partial occlusion handling," *Cybernetics, IEEE Transactions on*, vol. 43, no. 6, pp. 2005–2019, 2013.

[18] T. Zhang, B. Ghanem, S. Liu, C. Xu, and N. Ahuja, "Robust visual tracking via exclusive context modeling," *Cybernetics, IEEE Transactions on*, vol. 46, no. 1, pp. 51–63, 2016.

[19] Z. He, S. Yi, Y.-M. Cheung, X. You, and Y. Y. Tang, "Robust object tracking via key patch sparse representation," *Cybernetics, IEEE Transactions on*, 2016.

[20] B. Ma, L. Huang, J. Shen, and L. Shao, "Discriminative tracking using tensor pooling," *Cybernetics, IEEE Transactions on*, 2015.

[21] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 2544–2550, IEEE, 2010.

[22] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Computer Vision–ECCV 2012*, pp. 702–715, Springer, 2012.

[23] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 37, no. 3, pp. 583–596, 2015.

[24] J. Ding, Y. Huang, K. Huang, and T. Tan, "Robust object tracking via online learning of adaptive appearance manifold," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pp. 1863–1869, IEEE, 2011.

[25] B. Babenko, M.-H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 983–990, IEEE, 2009.

[26] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *Computer Vision–ECCV 2012*, pp. 864–877, Springer, 2012.

[27] M. D. Jenkins, P. Barrie, T. Buggy, and G. Morison, "An extended real-time compressive tracking method using weighted multi-frame cosine similarity metric," in *Education and Research Conference (EDERC), 2014 6th European Embedded Design in*, pp. 147–151, IEEE, 2014.

[28] M. Danelljan, F. S. Khan, M. Felsberg, and J. v. d. Weijer, "Adaptive color attributes for real-time visual tracking," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pp. 1090–1097, IEEE, 2014.

[29] I. Matthews, T. Ishikawa, and S. Baker, "The template update problem," *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 6, pp. 810–815, 2004.

[30] L. Sevilla-Lara and E. Learned-Miller, "Distribution fields for tracking," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 1910–1917, IEEE, 2012.

[31] X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai, "Minimum error bounded efficient l1 tracker with occlusion detection," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 1257–1264, IEEE, 2011.

[32] F. Pernici and A. Del Bimbo, "Object tracking by oversampling local features," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, no. 12, pp. 2538–2551, 2014.

[33] A. Zarezade, H. R. Rabiee, S. Member, A. Soltani-farani, and A. Khajenezhad, "Patchwise Joint Sparse Tracking with Occlusion Detection," vol. 23, no. 10, pp. 1–13, 2014.

[34] D. Deng, Q. Zhu, and J. Yan, "Compressive tracking via oversaturated sub-region classifiers," *IET Computer Vision*, vol. 7, no. October 2012, pp. 448–455, 2013.

[35] J. Zhang, S. Ma, and S. Sclaroff, "Meem: Robust tracking via multiple experts using entropy minimization," in *Computer Vision–ECCV 2014*, pp. 188–203, Springer, 2014.

[36] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 8, pp. 1619–1632, 2011.

[37] J. Ding, Y. Huang, K. Huang, and T. Tan, "Robust object tracking via online learning of adaptive appearance manifold," *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pp. 1863–1869, 2011.

[38] J. Ding, Y. Tang, W. Liu, Y. Huang, and K. Huang, "Tracking by local structural manifold learning in a new ssir particle filter," *Neurocomputing*, vol. 161, pp. 277–289, 2015.

[39] J. Burg, "Maximum entropy spectral analysis," Ph.D. Thesis, Stanford University, California, 1975.

[40] A. Doucet, N. De Freitas, and N. Gordon, *An introduction to sequential Monte Carlo methods*. Springer, 2001.

[41] A. Levey and M. Lindenbaum, "Sequential karhunen-loeve basis extraction and its application to images," *Image Processing, IEEE Transactions on*, vol. 9, no. 8, pp. 1371–1374, 2000.

[42] M. Brand, "Incremental singular value decomposition of uncertain data with missing values," in *Computer VisionECCV 2002*, pp. 707–720, Springer, 2002.

[43] P. Hall, D. Marshall, and R. Martin, "Adding and subtracting eigenspaces with eigenvalue decomposition and singular value decomposition," *Image and Vision Computing*, vol. 20, no. 13, pp. 1009–1016, 2002.

[44] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pp. 2411–2418, IEEE, 2013.

[45] X. Jia, H. Lu, and M.-H. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 1822–1829, IEEE, 2012.

[46] M. Danelljan, G. Häger, F. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *British Machine Vision Conference, Nottingham, September 1-5, 2014*, BMVA Press, 2014.

[47] K. Zhang, L. Zhang, and M.-H. Yang, "Fast compressive tracking," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, p. submitted, 2014.

[48] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust l1 tracker using accelerated proximal gradient approach," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 1830–1837, IEEE, 2012.

[49] S. Hare, A. Saffari, and P. H. Torr, "Struck: Structured output tracking with kernels," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 263–270, IEEE, 2011.

[50] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time object tracking via online discriminative feature selection," *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 4664–4677, 2013.

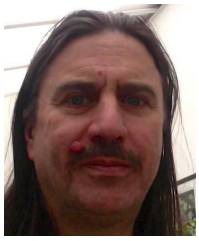
[51] S. He, Q. Yang, R. W. Lau, J. Wang, and M.-H. Yang, "Visual tracking via locality sensitive histograms," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pp. 2427–2434, IEEE, 2013.

[52] N. Wang and D.-Y. Yeung, "Learning a deep compact image representation for visual tracking," in *Advances in Neural Information Processing Systems*, pp. 809–817, 2013.

[53] Y. Wu, B. Ma, M. Yang, J. Zhang, and Y. Jia, "Metric learning based structural appearance model for robust visual tracking," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 24, no. 5, pp. 865–877, 2014.

[54] T. Yang, B. Li, and M. Q.-H. Meng, "Robust object tracking with reacquisition ability using online learned detector," *Cybernetics, IEEE Transactions on*, vol. 44, no. 11, pp. 2134–2142, 2014.

Mark David Jenkins received the B.Sc. Hons degree in Mechatronics at Glasgow Caledonian University, Glasgow, United Kingdom, in 2013. He is currently a Research Student at Glasgow Caledonian University working towards the Ph.D. degree in Visual Object Tracking. His research interests include image processing, pattern analysis, machine vision and machine learning.



Peter Barrie received the BSc (Hons) Degree in Computer Science, and an MSc (research) in Embedded Systems from the University of Strathclyde, Glasgow. He is currently a Senior Lecturer in Computing within the Department of Computer, Communications and Interactive Systems at Glasgow Caledonian University. His main research interests include Pervasive Systems and applications of Internet of Things.

Tom Buggy obtained his Ph.D. in Physics from the University of Glasgow in Scotland. He is currently professor of Computer Engineering at Glasgow Caledonian University and Head of the Department of Computer, Communications and Interactive Systems within its School of Engineering and Built Environment. From 2007 until 2011 Professor Buggy was Head of the Division of Communication, Network and Electronic Engineering and, from 2001 until 2007, Professor Buggy was also Head of the Division of Computing. Prior to joining Glasgow Caledonian University in 1996, Professor Buggy spent 14 years as a Principal Consultant at BAe Defence Systems. Professor Buggy research interests centre on: the application of quantitative and qualitative information processing and analysis techniques to emulate human intelligence; the development of computer systems that support human decision making, learning and problem solving; and the engineering of effective human computer interfaces and usable computer systems.

Gordon Morison received the BEng (Hons) Degree in Electronic and Electrical Engineering, and a Ph.D. in Signal Processing from the University of Strathclyde, Glasgow. He is currently a Lecturer in Signal and Image Processing within the Department of Engineering at Glasgow Caledonian University. His main research interests include Computer Vision, Pattern Recognition and Signal/Image Processing applied to biomedical applications.